September 1991 ISSN 0956-9979

VIRUS BULLETIN

THE AUTHORITATIVE INTERNATIONAL PUBLICATION
ON COMPUTER VIRUS PREVENTION,
RECOGNITION AND REMOVAL

Editor: **Edward Wilding**

Technical Editor: **Fridrik Skulason**

CONTENTS

EDITORIAL

Burger and his Apologists

As reported in last month's *VB*, Abacus, the Michigan based publishing company, released Ralf Burger's *Computer Viruses and Data Protection* on July 12th 1991. His original book, *Computer Viruses: A High Tech Disease* caused a wave of protest from the responsible research community for its inclusion of source code listing to a number of viruses including an annotated disassembly of the Vienna virus. *VB*'s technical editor estimates that Burger is reponsible (albeit by proxy) for the creation of more viruses than any other individual with the possible exceptions of the 'Dark Avenger' and the unkown author of the Jerusalem virus. Most bona fide researchers familiar with the Vienna virus and with the Virdem demonstration virus which Burger has actually marketed in Germany are united in the opinion that this man should be ostracised permanently.

However, in a recent interview (August 1991) published in a newsletter called *VNI*, Burger makes it clear that he wishes to 'return to the fold' - his stated reason being that he now requires virus binary and disassemblies in order to maintain his anti-virus software. This includes a program called *Virus Secure for Windows* which is also marketed by Burger's publisher *Abacus* in the United States.

In this interview Burger claims to regret publishing source code in his first book and states that he had not considered the possibility that hackers would use this information to assemble and distribute live virus code. This source code, he assures the sympathetic interviewer, would definitely *not* be included in future editions of the book. Nowhere is the date of this interview mentioned, so it is difficult to ascertain exactly when or where Burger made these statements. Superficially, these assurances seem convincing - the interviewer certainly appears to have believed Burger's sincerity describing him as 'genuinely remorseful'. The original book was published in 1987 when virus research was in its infancy and ethical questions had not arisen. Many notable researchers published articles and conducted studies in 1987 and 1988, which would obviously digress from the tacit ethical code to which genuine researchers now adhere. Suspicions abound and accusations still fly about a number of respected and eminent researchers accused of having assembled virus code in the past. The apologists for Burger's behaviour point to these isolated and bygone incidents inferring that they are directly comparable to Burger's continuing transgressions.

Two weeks prior to the publication of this interview, *Abacus* printed and released the first run of Burger's second edition, retitled *Computer Viruses and Data Protection*. Despite the aforementioned assurances from Burger, there in print were the very same source code listings that have caused so much chaos to computer users worldwide. In the rewritten preface to the book (dated June 1991) Burger writes:

> *"Some readers may believe we should not have included the program code and examples published in this book. The reason that we included these samples is to show how easy it is to write a computer virus. We felt that a good way to know learn* [sic] *how to avoid computer viruses, is to know how these programs work."*

It is therefore clear that Burger lied in the interview about republishing source code. To claim that users need to be conversant with assembly language to defend themselves against computer viruses is deliberately misleading. The source code was included in the book because it increases its value from near worthlessness to that of a useful 'do it yourself' cookbook for aspiring virus writers. Burger cannot plead ignorance - he knows only too well the damage and inconvenience that resulted from his first book; that he has wilfully re-published this information exposes the depth of this man's cynicism and greed.

It is possible that Burger will attend a meeting of the *European Institute for Anti-Virus Research* (*EICAR*) which will take place in Brussels later this month. He has stated his intention to attend and it is presumed that discussion about his readmittance to respectable research will arise. The dangers inherent in his readmission are manifest. Ultimately he has undermined the trust of computer users and researchers alike - the flow of information, disassemblies and binary code between researchers is likely to cease abruptly if there is any suspicion or possibility that Burger might be in receipt of such material - there is absolutely no guarantee that he will not use it to compose an even more explicit third edition of his odious book. Professor Klaus Brunnstein, president of *EICAR* has gone on record as stating that individuals involved in developing or publishing virus code will be permanently excluded from the legitimate research community - it is to be hoped this rule prevails and that the apologist lobby is sent packing.

Washburn's V2P7

Mark Washburn claims to have developed yet another 'research' virus, information about which he posted to *Compuserve* on 16th August. According to Washburn, the virus is capable of generating over 1 billion combinations. The algorithm used in his previous V2P6 virus is specific to the .COM file format; however, Washburn reports that the algorithm used in V2P7, although coded for .COM files can be extrapolated for .EXE format transfer while retaining its encrypting ability.

Further questions regarding V2P7 should be addressed to: *V2P7, c/o Mark A. Washburn, 4656 Polk Street NE Columbia Heights, MN 55421, USA.*

The only question which immediately springs to mind is why does Mr. Washburn persist with this nonsense?

TECHNICAL NOTES

FDISK and DOS 5.0

Further to Kevin Powis' letter (*VB*, August 1991, p. 5), readers should be aware that the *FDISK* utility supplied with DOS version 5.0 behaves in a slightly different way from previous versions. Removal of the boot marker 55AA from the Master Boot Sector of a fixed disk will cause *FDISK* to overwrite the sector. However *FDISK* will also delete the contents of the DOS Boot Sector, rendering all data on the disk inaccessible without extensive rebuilding or reformatting!

This method cannot thus be recommended as a reliable way of removing Master Boot Sector viruses. Users would be much better advised to use a reliable disk editor, either to locate and copy the original Master Boot Sector back into place, or to keep copies of their boot sectors on floppy disk and restore from them should they encounter a boot sector virus.

'Are You There' Conflicts

Jonathan Lettvin of *Lotus Development Corporation*, Cambridge, Massachusetts has pointed out that the IBM PC virus 'Are You There' calls (published in *VB*, May 1991, p. 7) conflict with *Novell's NetWare* program NET3.EXE. On examination of the *Novell* INT 21H service routine, Lettvin reports a jump table of 64 functions mapped to INT 21H functions B4H to F3H, as well as function 69H. This means that the list of viruses which respond to 'Are you there' calls - starting with the Datalock virus, through to and including the Frere Jacques virus - formally conflict with the *Novell* INT 21H function map. Detection programs which use these calls will therefore cause spurious *NetWare* behaviour.

Intelligent Scanning

The major problem with traditional virus scanners is the need to keep them constantly up-to-date, in order to detect new viruses as they appear. 'Intelligent' scanners are an attempt to solve this problem. These programs do not employ any virus-specific information, such as search patterns, but instead rely on a set of rules. A typical set of rules might be:

1. Any .COM file which starts with a JMP to a location no more than a few kilobytes from the end of the file, and which modifies the contents of the first few bytes later and transfers control back to address 100H in the initial code segment is probably infected by a virus.
2. Any program which intercepts INT 21H, function 4BH (Load/Execute), and then opens the file being executed in Read/Write mode probably contains a virus.
3. Any program which contains INT 13H or INT 26H calls is potentially dangerous.

Programs which check for such conditions have existed for a long time - an old, (and mostly useless) example is the *CHK4BOMB* program. Recently a second generation of intelligent scanning programs has appeared. Even though they are not perfect, they can achieve a detection rate of over 75 percent - implying that the programs have a corresponding chance of detecting any new virus, which virus specific scanners are unable to detect.

The problems with intelligent scannerss are twofold - false negatives and false positives. False negatives are to be expected, as such programs are not intended to be the ultimate solution to the problem of detecting viruses. False positives are a more serious problem - if a program produces too many incorrect warnings it will not be used, or it may suffer from the 'Cry Wolf!' syndrome - a warning is ignored when it is in fact genuine. An example of a program which may produce a false positive is FORMAT.COM. It is actually an .EXE file, which has been converted to a .COM file with a small loader appended to the end. This extra code might trigger on this file, unless special steps were taken to exclude FORMAT.COM. The third rule would also cause an intelligent scanner to trigger, which is only to be expected considering the program's function.

'Intelligent' scanners will probably be the subject of a comparative review in a later edition of *VB*, but currently they should only be considered as supplementary to traditional scanners - they are still far from being a replacement.

Selecting and Testing Virus Patterns

As readers of *VB* know, a list of virus search patterns is published each month. Several factors must be considered when selecting a pattern. A brief description of the selection and verification process follows.

The first step is to determine whether a suitable pattern can be extracted. In the case of most self-modifying viruses no pattern can be found - publishing a pattern which only detects one instance of the virus is useless. This step is performed in two ways - by a limited disassembly of the virus, and by infecting a fixed collection of 12 'victim' files. This will also reveal some properties of the virus - whether it is memory-resident, has a trigger routine etc., and what types of files (COM and/or EXE or boot sectors) it infects.

If the virus is identical in the newly created sample files, the search for a pattern starts, subject to the following rules:

1. The pattern should not contain a sequence of code which is likely to be found in any non-related program. (This is nearly impossible in the case of high-level-language viruses, such as Kamikaze, unless rule 2 is broken).
2. The pattern should not contain references to any addresses outside itself, as they may become invalid if a new variant is created.

3. The pattern should preferably not contain instructions which exist in two different forms (such as XOR AX,AX), as it might then become invalid if a different assembler is used to assemble the source.

4. The pattern should not contain any data or text strings. (Consider for example that at least seven variants exist of the New Zealand virus, with only the text message changed).

5. The pattern should be unlikely to change in different versions of the virus, so the same pattern can be used to detect multiple viruses in the same family. (The HIV pattern is a good example of this).

It is often not possible to follow all the rules, and some of them have occasionally to be bent slightly or even broken.

When a suitable pattern has been found, it is verified by using it as input to a 'search engine' - which should then detect all the files which were infected in step 1. The pattern is then copied from one file to another, and sent to the *VB* editorial office by e-mail.

The ASCII formatted text is then imported directly into *PageMaker* for typesetting which minimises the chances of errors creeping in as might be the case using a facsimile message. The patterns are also tested using a search engine maintained by the editor which is run against each live virus sample received.

Before printing, the patterns are proof-read (by three people working independently) against the original e-mail dump.

Admittedly, the method described here is not 100 percent foolproof - testing for false positives (particularly in the case of High Level Language viruses) has not yet been included in the selection process. It should also be stated that not all patterns published by *VB* have been selected by the process described above. The original policy was to publish a specific pattern for each variant, so for example Fu Manchu and Jerusalem have separate patterns despite the fact that it would have been easy to find a common search pattern.

Different Scanning Speeds

One of the factors which can determine the selection of a virus scanner is its speed. Several scanners offer options to perform scanning at different levels of security, where the speed decreases as the security increases.

This provides several benefits - somebody in a low-risk environment might prefer the fastest option, perhaps using the slowest (and most secure) option occasionally. Somebody responsible for installing programs on a large network would probably want to use the most secure method on all new programs, just to be as sure as possible that the programs being installed are 'clean'.

Different levels of security can be implemented in a number of ways, including the following:

➤ The developer can incorporate a number of different search patterns for each virus. Using multiple patterns for the same virus improves the chances of detecting new variants of the original virus.

➤ The developer can use 'fuzzy matching'. A program which does not require accurate matching, but allows a few bytes to be different has an improved chance of detecting new variants. *IBM's VIRSCAN* is a notable example of a scanner employing this tactic. The biggest problem with this method is not the decrease in speed, but rather the increased chances of false positives. (See Mark Drew's letter on page 11.)

➤ The developer can elect to search entire files rather than areas where the virus code is likely to be located. With a few exceptions, parasitic viruses are either found at the beginning or the end of infected files. Normally there is no need to scan more than a small portion of the file, but scanning entire files may improve the chances of detecting variants which have been modified so that their length is significantly different from the original virus. (NTKC, a 23693 byte variant of the 648 byte Vienna virus is the best example of this.)

➤ The developer can ignore 'extinct' or 'laboratory only' viruses. This approach (known as 'selectivity') was discussed in the last two editions of *VB*. As the overwhelming majority of computer viruses are unknown in the wild or isolated within a limited geographical area, a considerable performance increase can be obtained by only searching for viruses known to be an actual threat. The question of just which viruses to exclude is a tricky one, however.

➤ The developer can limit his search to fixed locations. By searching only for each search pattern at one fixed location in each file, a tremendous increase in speed can be obtained, but this approach has a serious drawback - each variant may require a separate entry in the database and the method is unlikely to catch new variants which are created by modifying older ones.

The 'Mutation Engine'

In March 1991, the 'Dark Avenger' posted a note on *Fidonet*, announcing a soon-to-be released 'mutation engine' - a skeleton for constructing self-modifying encrypted viruses (*VB*, April 1991, p. 19). This program has now materialised (in source code form), and is being analysed.

The source code is well commented (in Bulgarian) and an English translation will be available soon. We hope to report on the implications of this development in detail in the next edition of *VB*.

OVERVIEW

Fridrik Skulason

July 1989 - July 1991: Significant Developments

When the first edition of *Virus Bulletin* was published in July 1989 the computer virus situation was quite different from what we see today.

The major change is in the number of viruses - back then, *VB* reported a total of 12 MS-DOS viruses, which could be described on a single page, with room to spare. However, the last full listing (*VB*, July 1991) occupied a total of 22 pages and included 321 separate search patterns!

Figure 1 illustrates this increase with the line graph representing the increase in the number of identification strings published in *VB* during the period July 1989 to July 1991.

For the dedicated statistician, the exact figures are:

Jul 1989	12
Aug	22
Sep	24
Oct	25
Nov	26
Dec	44
Jan 1990	58
Feb	60
Mar	77
Apr	84
May	93
Jun	102
Jul	113
Aug	109
Sep	114
Oct	122
Nov	135
Dec	140
Jan 1991	226
Feb	243
Mar	248
Apr	270
May	282
Jun	300
Jul	321

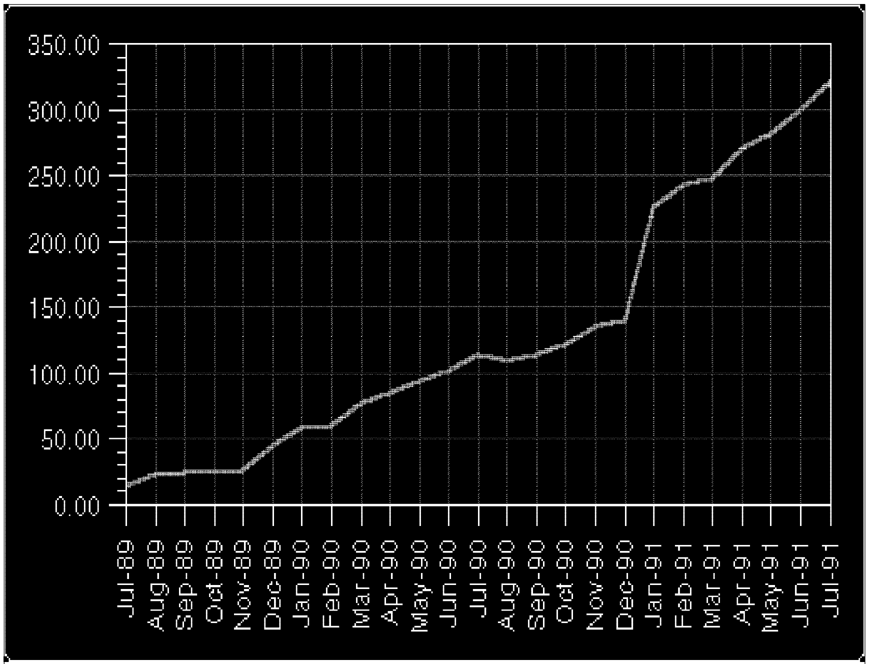


Figure 1. The inexorable increase in IBM PC virus search patterns over a 24 month period. The inevitable consequence will be ‘scanner exhaustion’.

Note that the slight drop in August 1990 is due to a revision at that time in the formulation of the *VB Table of Known IBM PC Viruses*. The number of patterns published by August 1991 was 341, which reaffirms the linear trend shown in *Figure 1*.

The large and anomalous jump in January 1991 is intriguing, but it has a simple explanation, as numerous viruses from Eastern Europe first became available to researchers in the West at a conference in Hamburg in December 1990. Most of them were several months old by then and if the graph is adjusted for this, it is evident that it does not show an exponential curve, but rather one which approaches a straight line, with 12-13 new search patterns being extracted each month on the average. So why is the number of viruses often said to be growing at an exponential rate?

The reason for this inconsistency is that the graph shows the number of virus *identification strings*, but the number of *variants* detected by each string has been constantly increasing. A graph of the number of known virus variants does indeed exhibit a more exponential-looking curve. Unfortunately exact statistics about the appearance of minor variants are not available - this is explained by the fact that no researcher predicted this development at the time when statistics started to be maintained. The number of Macintosh virus specimens has not grown at a similar rate.

‘Real World’ Outbreaks

The number of virus specimens per se is of academic interest - of more importance are the number of genuine virus attacks occurring in the real world and the level of disruption that this is causing to users. Without reliable statistics, estimating the penalties imposed by computer virus infections (in terms of man-hours of systems denial and recovery costs) is an impossible task. Anti-virus software houses in Europe and the United States are reporting a substantial increase in the number of genuine attacks (the number of these reports has increased noticeably over the last six months) but accurate data on this is hard to come by. The report published this month (see pages 13-15) about virus penetration in the United Kingdom provides some insight into the extent of the problem. *IBM's T. J. Watson Research Center* in New York estimates that approximately 10 percent of virus specimens have been identified in the wild in the United States.

A Vicious Circle

Although interesting, the explosion in the number of viruses is not the only noteworthy phenomenon to occur in the virus field over the past couple of years. The viruses have changed, virus writers have adopted new techniques, but so have the developers of anti-virus tools. The interaction between those two groups has been discussed before - new viruses use new techniques, which are countered by new anti-virus programs, which results in the creation of a new generation of viruses, and so on in what might best be described as a vicious circle.

Eastern Europe

The viruses listed in the first edition of *VB* were: 405, Brain, Cascade, Datacrime, Fu Manchu, Italian, Jerusalem, New Zealand, Pentagon, Traceback, Vienna and Yale. Some of them, such as Jerusalem and New Zealand are still considered very common, despite constant efforts during the past few years to eradicate them. Others, such as 405 and Datacrime only caused a few isolated infections, and Pentagon was never available in working form. Those twelve viruses originated in Western Europe, USA, Pakistan, Israel, New Zealand and the Philippines, but in December 1989 *VB* listed the first virus from Eastern Europe. It was soon followed by more, but the outpouring of viruses from Eastern Europe took the anti-virus community by surprise. Fortunately few other such unforeseen developments have occurred.

Perhaps the only other real surprise was the 'minimalist' movement - the trend toward the creation of the smallest functional virus. It started with the 163 byte Danish Tiny virus, but the current record holders are Micro-128 (resident) and Minimal-30 (non-resident).

Increased Programming Sophistication

In addition to the 'minimal' viruses, we have also seen a trend toward more sophisticated viruses, culminating in samples such as Whale. Sophistication is not synonymous with success, and the Whale virus must be considered a failure in most respects - it is much too bulky and obvious. The only real effect of the virus was to keep many virus researchers occupied for quite a while in the process of its disassembly.

The most interesting (and perplexing) technical developments over the past two years are self-modifying encryption, as used by V2P2 and V2P6, (*VB*, March 1990) and stealth viruses (*VB*, June 1991).

Self-modifying encryption is a good example of an 'efficient' new technique, as demonstrated by the fact that many anti-virus programs are not able to detect all the self-modifying viruses in all instances. The reason for this is that instead of a simple search string, an algorithmic method must be used, and if this is not tested sufficiently it may fail to detect some infected files. In a recent test by competitor S&S Ltd., *Symantec Corporation's Norton Anti-Virus* (which claims to detect the Tequila virus) failed to identify 62 out of 63 Tequila infected files.

The techniques used by the self-modifying viruses range from the very simple ones, where the difference in the decryption program between two infected files may be as small as two bytes to the complex methods used by the Whale virus. Fortunately only a small fraction of all viruses use self-modifying encryption, but the latest development - the so called 'mutation engine' developed by the self-styled 'Dark Avenger' may well result in a sharp increase in their numbers in the near future.

Stealth viruses actually predate the two year period covered by this article, as the early MS-DOS virus Brain used stealth methods, but their use in parasitic (program) viruses is a recent invention. Fortunately the complexity involved in the development of stealth viruses is beyond the ability of most virus writers while those programmers capable of producing a competent stealth virus from scratch are likely to be gainfully employed writing useful software instead. At the time of writing approximately 3 percent of known PC viruses employ stealth tactics.

Subversion

In March 1990, *VB* contained an article on the subject of the new generation of computer viruses - describing the two techniques just mentioned, as well as a series of attempts to subvert specific anti-virus tools, methods for bypassing interrupt monitoring programs and the growing concern over the availability of virus source code.

In the eighteen months since then the appearance of new virus writing methods has actually declined. Recent viruses have used new loopholes, and two new techniques have been introduced - hybrid 'multi-partite' viruses and 'companion' viruses - both developments were predicted and are of relatively minor importance.

Many of the new techniques adopted by virus writers during the past two years are in response to the increased circulation and usage of anti-virus tools. Self-modifying encryption and stealth methods are a defensive reaction against virus scanners, but interrupt 'stripping' (discussed later in this article) is a response to generic monitoring programs.

The most direct response is naturally to make the virus aware of a particular anti-virus program. A few viruses employ this feature, the most recent example being the Rage virus, which searches for *Microcom Software Division's Virex-PC* in memory. If it is found, the virus simply returns control to the host program.

'VXs'

Virus distribution has also changed in the past two years. Occasionally researchers still receive viruses directly from the authors, but the virus exchange BBSs (now commonly referred to as 'VXs') have actually started to serve as collection and distribution points - new viruses accumulate there, and are occasionally downloaded and sent to members of the virus research community. Uploading a virus to a 'VX' usually results in its immediate recognition and thus it is not a particularly efficient way to spread it. However, the uncontrolled dissemination of both binary and source code by VXs in Europe and the United States is a matter of serious concern to genuine researchers.

There have been a number of cases of mass distribution of live virus code, for example infected cover disks distributed with computer magazines, but those cases have fortunately been few. Instances of contaminated commercial or 'shrink-wrapped' software are also on the increase due to poor quality control either at the development stage or during disk duplication - lending strength to the argument that no software, whatever its source, can be trusted implicitly.

Dynamic Decompression

A year ago the first *LZEXE*-packed viruses appeared, and their frequency has been increasing steadily. This has forced anti-virus authors to respond - some scanners are now able to scan compressed files, whereas others only report that the files are packed, and cannot be scanned reliably.

The problem with *LZEXE*-packed files (or files packed by similar programs such as *ICE*, *DIET*, *PKLITE* or *EXEPACK*) is that unless a virus scanner recognises packed files, it will not detect first-generation infected files, only later generations. *VB* has received a number of reports of recurring virus contamination caused by packed virus-infected files which have remained undetected on disk.

Naming Conventions

The challenges facing the developer of anti-virus programs have changed in several other ways as well. In the pages of the first few editions of *VB* one can find a discussion of two problems which are of rather less concern today.

The first regards the naming of viruses - should they be given names, or just a number, indicating their infective length? This problem was initially resolved in favour of names, but now the numbers are making a comeback - the increased number of virus variants has resulted in viruses often receiving a name combining two parts, a family code and a length identifier, such as 'Murphy-1173'.

Research Disputes

The second problem in 1989 was the reluctance of some virus researchers to share their virus collections. This problem still persists, as illustrated by the fact that a virus researcher in the US recently received a large batch of viruses from a well-known UK researcher, on condition that the viruses would not be sent to another well-known UK researcher. Virus exchange between researchers is still based on trust. There have been various calls for the implementation of a code of ethics - unfortunately no-one seems to agree about even the basic tenets of such a proposed code. Despite occasional feuding, virus distribution among recognised members of the anti-virus community is generally reasonably fast and problem-free.

Redundant Technologies

However, as old problems vanish, new ones appear. Developments over the past couple of years have made some types of anti-virus programs obsolete, although their suppliers have not always realised this. An obvious example of a redundant technology is one that was pioneered by *Virus Bulletin* itself - the offsets of virus identification strings. Using the offsets made sense back in 1988 or 1989 when each string was only used to identify one particular virus variant. This is no longer true and as the offsets published in *VB* were considered to be of no practical use to anybody, the decision was made to drop them altogether.

The use of signature offsets is not the only technology which has become outdated - the same applies to many generic interrupt monitoring programs. These programs intercept various interrupt functions, and watch for suspicious activity, such as a program formatting the hard disk or modifying an executable program. The sad fact is that many of the available programs are simply not able to perform as advertised. Recent viruses are able to bypass them altogether, which also applies to some hardware products (see the review of *Knoxcard, VB,* July 1991, pp. 38-40).

Despite this, some programs are still on the market and advertised as 'able to prevent any virus from damaging your data', but unfortunately those claims are generally not true. It must be noted that some monitoring programs are better than others, but a comparative review has not yet been scheduled.

The reason for viruses being able to bypass most monitoring programs is the existence of various loopholes which enable virus programs to obtain the original entry points of the interrupt functions. So far three methods have been used. The most primitive method is that used by the December 24th virus, which uses an undocumented interrupt function to obtain the segment address of DOS. It then performs a string search for the first few instructions of the original INT 21H function in several versions of DOS. This simple method has a drawback, from the virus writer's point of view - the virus might not be functional under a new version of DOS, such as DR-DOS 5.

A more direct approach is taken by some viruses which use an undocumented, but well-known function to obtain the original INT 13H entry point, which will for example disable most software attempts to protect the hard disk from formatting.

The most sophisticated method is sometimes known as ‘stripping’ - it basically involves setting the single-step bit, and then tracing through an interrupt call, until the segment part of the address of the current instruction is considered to be ‘right’, for example if the address is above F000:0000, when tracing INT 13H.

Some generic monitors are also a bit out of date in other respects - for example they may fail to detect when a program ‘goes TSR’. The traditional way to do this is to use the standard DOS methods - either INT 27H or INT 21H, function 31H. Both methods are easily detectable by a generic monitoring system, so other methods have been adopted in the past two years. One is simply not to allocate the memory at all - just copy the virus to some area high in RAM and hope it does not get overwritten by another program. Another method is to create a ‘hole’ at the top of RAM, either by directly changing the size of the last memory block or by lowering the value stored at 40:13. Those changes could be detected by any monitoring program, so an even more sophisticated method was developed, which is used by viruses such as Micro-128. It involves copying the virus to the upper half of the Interrupt Table, an unused disk buffer or another ‘unused’ area.

One completely discredited technology which has long since fallen from grace is the concept of inoculation software - in 1989 a number of primitive software packages contained inoculation routines designed to protect programs and system files by introducing viral self-recognition ‘signatures’ or ‘are you there’ calls. (These signatures are employed by virus writers to prevent multiple reinfections of a file.) Unfortunately, the multitude of virus self-recognition signatures now in existence renders this approach completely useless for general defence. Inoculation routines can usually only protect against a single virus or related subset at any one time.

Scanner Exhaustion

Virus-specific software is suffering from a different problem - the unremitting increase in the number of virus variants is causing a corresponding growth in their databases - generally degrading the speed of these programs. The intelligent scanners, discussed in the technical notes on page 3, have been developed as an attempt to restore scanner ‘run-times’ and diminish scanner maintenance - generally speaking they do not have to be modified when a new virus appears. Some manufacturers are exploiting the ‘virus numbers’ issue - claims that Scanner ‘X’ detects 700 viruses while Scanner ‘Y’ detects only 600 should be dismissed as meaningless marketing hype. Interestingly, many reputable manufacturers now advocate the use of at least two scanners from different sources - a development which would have been unthinkable only twelve months ago.

Virus ‘Vapourware’

The October 1989 edition contained the following note on virus ‘vapourware’:

> ‘*Another disturbing development is the repeated claim that some of the viruses in the Reported Only section of the Known IBM PC Virus Table do not in fact exist.’*

The viruses in question were 2730, Agiplan, Dbase, Missouri, Mistake, Nichols, Oropax, Screen and Swap. Agiplan, Dbase, Mistake, Oropax and Swap appeared later, although the reappearance of Agiplan is surrounded by mystery. Missouri, Nichols and Screen are still listed in the Reported Only section, which leaves only 2730. It appears never to have existed at all, except as a ‘phony’ entry in a list of virus identification strings designed so that the originator could identify competitor’s products which incorporated his search data into their scanners. Today it appears that hexadecimal search patterns are (generally) no longer regarded as intellectual property subject to copyright.

The Media

The treatment of viruses in the media has also changed in the past two years. After the October 13th 1989 Datacrime media fiasco (which induced something akin to mass hysteria in The Netherlands), viruses were ignored for a while, and even dismissed as mythical. This appears to be changing; viruses are now perceived more realistically as a real problem, which will neither disappear nor cause any massive catastrophe in the immediate future.

Trade press attention, has switched from the virus code itself to the performance of anti-virus software, (particularly the relative merits of virus-specific scanners). This tendency is reflected in the technical journals and newsletters dedicated to reporting the computer virus phenomenon - *VB*, *S&S Ltd.* and the *National Computer Security Center* in the United States have conducted comparative reviews of software in the last six months. Needless to say, accusations of bias arise with every publication of a comparative review - such are the commercial stakes involved.

Research Initiatives

The research and coordination effort has also accelerated to keep pace with developments, perhaps the most significant initiatives to have occurred during the last six months are the *European Institute for Computer Anti-Virus Research* (*University of Hamburg*), *The National Institute of Standards & Technology* government-industry consortium to combat the virus threat (Gaithersburg, Maryland, USA) and the *Computer Virus Strategy Group* (*Metropolitan Police Computer Crimes Unit*, London, UK).

It is to be hoped that a more united anti-virus community will result from these (and other) initiatives.

IBM PC VIRUSES (UPDATE)

Updates and amendments to the *Virus Bulletin Table of Known IBM PC Viruses* as of 21st August 1991. Hexadecimal patterns may be used to detect the presence of the virus with a disk utility program, or preferably a dedicated virus scanner.

Type Codes

C = Infects COM files **E** = Infects EXE files **D** = Infects DOS Boot Sector (logical sector 0 on disk)
M = Infects Master Boot Sector (Track 0, Head 0, Sector 1) **N** = Not memory-resident after infection
R = Memory-resident after infection **P** = Companion virus

Seen Viruses

Brunswick, Stoned 16 - MR: Detected in the wild in the US. Infects first fixed drive and floppy drives A and B. On floppy disks the original boot sector is stored in Head 1 Track 0 Sector 3 and may cause directory corruption. On hard disks the original boot sector is stored in Head 0, Track 0 Sector 16.

```
Brunswick       D4FF E8E7 FF74 252E C606 2901 00B8 0103
```

Delyrium-1638, Move - CER: A ‘Cracker Jack’ virus detected by the HIV pattern.

Interceptor-Vienna - CN: This mutation written by ‘Cracker Jack’ is quite similar to the Monxla-B mutation. The search pattern can also be found in Monxla-B, but the viruses can be distinguished by different lengths.

```
Interceptor     B903 008B D683 C20D CD21 8B54 068B 4C04
```

Lao Doung - ?: Sample just received and awaiting analysis. A boot sector virus from Thailand. It reportedly plays a Laotian funeral dirge when it activates.

```
Lao Doung       A34C 0006 1FF6 C280 7539 BB00 7EBA 8001
```

Locker - CER: A 1642 byte mutation of the Murphy virus, written by Cracker Jack and detected by the HIV pattern. The virus has not been fully analysed yet, but under certain circumstances it will ask the user for a password. The first generation sample was packed with the ICE program, so it will not be found with the HIV signature.

Michelangelo - MR: A mutation of the New Zealand virus, which will activate on March 6th and format the hard disk.

```
Michelangelo    BE00 7C33 FFFC F3A4 2EFF 2E03 7C33 C08E
```

Minimal-46 - CN: A primitive overwriting virus which does nothing but replicate.

```
Minimal-46      D8BA 0001 B12E B440 CD21 B43E CD21 B44F
```

New BadGuy, Milan Overwriting-208, Crackpot-208 - CN: A 208 byte mutation of the BadGuy virus by ‘Cracker Jack’, created by adding NOP instructions at various locations around the code. The only effect other than replication is to display a message on Mondays.

```
New BadGuy      2E8A 1780 F243 90B4 02CD 2190 43FE C990
```

NoInt - MR: Boot virus with no payload which infects floppies in A and B as well as the hard disk. Infects when disk read is attempted, and returns the original boot sector when Sector 1 is read. The original boot sector is stored in Head 1 Track 0 Sector 3 on diskettes and Head 0 Track 0 Sector 7 on hard disks.

```
NoInt           00B9 0002 161F 33F6 8BFE FCF3 A436 FF2E
```

Possessed - CER: A 2438 byte virus (with a 2446 byte variant reported), which contains the text ‘POSSESSED! Bwa! ha! ha! ha! ha! Author: JonJon Gumba of AdU’. The virus is reported to delete files occasionally, after it has been resident for a while.

```
Possessed       8BF2 83C6 028B DE80 3C5C 7506 8BDE 43EB
```

Rape - CR: Two viruses with the same primary effect of overwriting the first 256 sectors of each drive. The shorter is 500 bytes long, but the longer one, which is 747 bytes has limited stealth-like abilities, as no increase in file length is visible if the DIR command is given while the virus is active in memory.

```
Rape            B980 00AC 3C61 7206 3C7A 7702 2C20 8844
```

Revenge Attacker - CR: This virus produces a strange effect on some machines, as directories may appear corrupted, containing multiple copies of the same file. The major effect of this virus is the destruction of all files on the disk. It is 1127 bytes long, and reported to have originated in the Philippines.

```
Revenge Attack  7510 4080 3F00 750A 4080 3F00 7504 F8E9
```

Thursday 12th - CER: An encrypted virus from Germany which triggers every Thursday 12th, popping up a window with a warning that the next day is Friday 13th. Calls itself VirCheck V1.2 (C)1991. Text includes ‘acknowledgements’ to Patricia Hoffman and John McAfee. Avoids infecting any files matching patterns ‘SCAN’, ‘CLEAN’, ‘VIR’, ‘ARJ’, ‘FLU’, ‘COMMAND’.

```
Thurs 12th  83F9 0074 0951 5630 2446 E2FB 5E59
C39C
```

Traveller - CER: Reported at large in the USA, 5th August 1991 (RG Software). A 1220 byte virus which infects COM (including COMMAND.COM) and EXE files. Infection is via Function 4BH (LOAD AND EXECUTE) and Function 36H (GET FREE SPACE). When a LOAD AND EXECUTE call is issued, a program and one other file in current directory are infected. When GET FREE SPACE request is issued (e.g. by the DIR command) one file in current directory is infected. Infection marker is the seconds field set to 62 and COM files will increase in size by 1220 bytes and EXE files by 1237 to 1251 bytes. The message

```
!!!!!!!—>> Traveller (C) BUPT 1991.4 Don’t panic
I’m harmless <<—!!!!!!!
```

flashes bright and dim green on blue background on line 13 of the screen after 23 infections.

```
Travel A303 0029 1612 00A1 1200 8EC0 0E1F 8BDE
```

Twin - ERP:Companion virus with no payload. Likely to crash where infected file larger than about 64K

```
Twin   B810 FFCD 213C 0775 07E8 2500 B44C CD21
```

Virdem-Killer - CN: This mutation is closely related to the original Virdem virus. The length is unchanged at 1336 bytes, although some text strings have been altered. The virus is detected by the Virdem pattern.

Xabaras - CER: An encrypted, overwriting 1972 byte virus. It is written by ‘Cracker Jack’. A mutation of the Leprosy virus.

```
Xabaras   908A 2790 9090 9090 9090 3226 0601
9090
```

Reported Only

923, Hey You - CER: Samples of the virus have not been found to replicate.

Adidas, Elephant, Pink Elephant - ?: Samples either appear to do nothing or just display the message ‘My Adidas!’ when a program is run. There is no evidence yet that the programs are virulent, even though they have been reported as such elsewhere.

Burghofer - CR: A 525 byte virus from Switzerland.

RMIT - CN: A mutation of Leprosy, 666 bytes long.

Spanish April Fools - ER: A 1417 byte virus from Spain. The proposed name of the virus is rather odd, as it is reported to activate on December 28th of any year, interfering with various commands such as COPY or REN.

VIRUS BULLETIN

EDUCATION, TRAINING AND AWARENESS PRESENTATIONS

Education training and awareness are essential as part of an integrated campaign to minimise the threat of computer viruses and Trojan horses.

Virus Bulletin has prepared a presentation designed to inform users and/or line management about this threat and the measures necessary to minimise it. The standard presentation consists of a ninety minute lecture supported by 35mm slides, followed by a question and answer session.

Throughout the presentation, technical jargon is kept to a minimum and key concepts are explained in accurate but easily understood language. However, a familiarity with basic MS-DOS functions is assumed. The presentation can be tailored to comply with individual company requirements and ranges from a basic introduction to the subject (suitable for relatively inexperienced users) to a more detailed examination of technical developments and available countermeasures (suitable for MIS departments).

The aim of the basic course is to increase user awareness about computer viruses and other malicious software without inducing counterproductive ‘paranoia’. The threat is explained in comprehensible terms and straightforward, proven and easily-implemented countermeasures are demonstrated. An advanced course, aimed at line management and DP staff, outlines varying procedural and software approaches to virus prevention, detection and recovery.

The presentations are offered free of charge except for reimbursement of travel and any accommodation expenses incurred. Information is available from the editor, *Virus Bulletin*, UK. Tel 0235 555139.

LETTERS

Sir,

Following on from recent correspondence and *VB's* advice to users of *IBM's VIRSCAN* to add signatures from your publication to an addenda list there have been some reports of false positives. This is because *IBM's VIRSCAN* has an in-built mutation detection capability. The *IBM* signatures are prepared specifically with this capability in mind and are then extensively tested for false alarms. Users adding untested signatures risk false positives. This in no way criticises the effectiveness of the signatures or their positive contribution as an addition to any scanner.

IBM's *VIRSCAN* gives false positives when used with five of the July/August *Virus Bulletin* signatures.

Four of these signatures *can* be used if the 'no mutants' option is used, these were:

Cascade YAP

Fingers 08/15

Jerusalem Clipper

Trilogy

This means if no mismatched bytes are allowed then the user is probably reasonably safe from false alarms.

The other false positive was with the ZK-900 signature for which there is a signature in *VIRSCAN* version 2.1.2.

My general advice to anyone using the *VB* signatures as add-ons is that users should either test very thoroughly or use the 'no mutants' key phrase, then the testing requirement is largely eliminated but is none the less still advisable.

With regard to the correspondence on 'selectivity' it was interesting to see within your editorial that you warned UK readers to ensure that any virus-specific tool should detect Spanish Telecom, Tequila, and 2100 because they were in the wild in the UK.

The 'common virus issue' is one that we have discussed in closed forums inside *IBM* for some time now. My understanding of Dave Chess's research on common viruses was that it was aimed at understanding the demography of the 'wild' viruses. In fact he presented and released a paper for the *DPMA* earlier this year on common viruses. [*Proceedings, Fourth Annual Computer Virus & Security Conference*, New York, 14-15th March 1991.]

My interest is in providing the most appropriate advice to customers while Dave's interest is more research based - both of us have the same objective of understanding virus spread patterns. This will help in advising users on better processes to avoid the risk of virus infections.

Your editorial advice on the need to protect against the high risk in the wild sub-set of viruses makes a case in support of the point made by Dave Chess in his letter in the August edition for '...give the reader some idea of how a product performed against the most important sub-set of their (the publication's) complete test-set'. This will not only qualify the effectiveness of a particular tool but add significant value in allowing a user to assess the technical value of a tool independent of the number of viruses the vendor claims for the product.

Mark Drew
Security Consultant
Systems Management Services Centre,
IBM UK

[*The importance of invoking the 'no-mutants' option when using VB search patterns within VIRSCAN has been discussed before (VB, July 1991, p. 10) but is worth re-emphasising. Concerning viruses in the wild, current tests (the results of which we hope to publish next month) have revealed a woeful failure to detect samples of Tequila and Spanish Telecom among a number of high-profile scanning programs. Ed*.]

Dear Mr. Wilding,

You may get some flak from the statement that *NetWare* 'attributes offer no protection against viruses' (*GPI Mystery Unravelled, VB*, August 1991, p.3), after all, execute-only is an attribute.

However, even in the case of execute-only the statement is substantially true. Arguments to the contrary would be similar to the first virus investigator, upon finding a virus which could not manipulate the read attribute, advocating setting file attributes to read-only as a technique for obtaining substantial protection.

The weakness does not lie in the ability to remove the attribute - you can't without erasing the file - but in the ability of the file-server to know if the workstation is really reading the file for the purpose of executing it. The mode of attack is clear: find out if delete and write rights are possessed, read the file surreptitiously, delete it, write it back modified. Execute-only is still a nice guy flag, unlike the rights flags which are designed for security purposes.

Sincerely,

G. Eric Babcock
Corporate Security Dept.
Novell Inc. Provo, Utah, USA.

Sir,

I must take issue with Kevin Powis (*Letters*, *VB*, August 1991) concerning his comments about Dr. Keith Jackson’s reviews. Rather than ‘aggressive’, I have often felt that Dr. Jackson’s observations were far too gentle considering the range and efficiency of the software/hardware that he has reviewed.

A reviewer’s task is to present to the non-specialist user a coherent picture of the product under consideration. Does it do the job it is supposed to do? Is it easy to use? Does it live happily within the multitude of different computing environments? Is it easy to install? Is the documentation informative and easy to access? Is it well supported? Many of these questions can only be answered subjectively and a good reviewer will make that quite clear. On the other hand, the negative aspects of a product must also be scrutinised.

I think credit is due to Dr. Jackson that he persevered in trying to ascertain the effectiveness of the *Knoxcard* (*VB*, July 1991, pp. 38-40) even when it could not be persuaded to function beyond the installation stage. It is surely up to the vendor to ensure that a product sent for review is absolutely the best that he has to offer! In an industry where genuine, independent technical expertise is in extremely short supply, I applaud Dr. Jackson’s efforts and look forward to many more honest and accurate reviews.

Mr. Powis also states “surely it is in the interests of all computer users to encourage any attempts to stem the tide of virus activity”. Applying this tenet to reviews of anti-virus software would surely stifle any negative reports on the grounds that the product was at least ‘an attempt’ at stemming the tide. A poor product may be worse than no product if it is inconvenient to use and engenders a false sense of security. Surely, where anti-virus products are concerned, it is in the interests of all computer users that reviewers should identify the strengths and weaknesses of individual packages. If a vendor wishes to gain better reviews, he has only to take notice of the reviewer and improve the product. Temporarily disregarding published reviews, if a customer is let down by a particular product he will certainly tell others and over a period of time it will become generally known which products provide a reliable service and which do not. An honest and qualified reviewer accelerates this process while a dishonest or incompetent reviewer impedes it.

I might add that I took Dr. Jackson’s suggestion that the *Knoxcard* code should be given to me for reverse engineering as a compliment (albeit tongue-in-cheek) - not as an instruction! Mr. K. Suresh insists that knowledge of the contents of specific locations/addresses is prerequisite for program subversion and this highlights exactly the point that Dr. Jackson was making - such specific knowledge is *not* a necessity and the vendor should be aware of this. In making such an assertion, Mr. Suresh reveals that he does not appreciate the full extent of the problem.

As a regular contributor to *Virus Bulletin*, I know a little of the tremendous effort that goes into its production. I am also aware of the care with which technical information is vetted to ensure that maximum help is given to users while revealing nothing to the virus writers. Mr. Powis is right when he says that *VB* has earned its position in the anti-virus community - its honest reviews and balanced approach enhance that position with every edition.

Your sincerely,

Jim Bates

Sir,

As part of our program of improving *VET* (*VB*, May 1991, pp. 18-19) I recently looked through *VB* to try and get an idea of the performance of other scanning software. However, I soon realised that the information on scanning performance given in your reviews was almost totally useless. A particularly bad example occurs in the August issue, in the review of *ProScan* (*VB*, August 1991, p.25) . This gives four different times, but I can find no mention of the type of PC, or drive, or the number, size and arrangement of the test files. Timing comparisons which do not test all programs on the same set of files on the same PC are of very little value. If you want your comparisons to be meaningful you should set aside a disk on a particular PC specifically for doing comparative tests. The disk should not be used for anything else as minor changes to the type, number, or arrangement of files could make a substantial difference to the results.

With best wishes,

Roger Riordan
Cybec Pty Ltd. Victoria, Australia.

Editor’s Reply:

For an explanation of the entries published in the table of results for ProScan, I would direct Mr. Riordan to the evaluation protocol published in VB, April 1991, pp. 6-7 - it actually says this in the review of ProScan (page 25). The information required is contained therein. We would publish this test information every month but consider such a level of repetition would prove irksome to the readership. Tests are ***always*** *conducted against this same machine and file configuration unless a specific statement appears to the contrary. The very notion that any evaluator would* ***ever*** *conduct time-tests on a variety of scanners against* ***different*** *machine and file configurations has been received with stunned incredulity! To quote the test protocol ‘All comparative tests should be conducted on the same machine with exactly the same file configuration.’ The scan speeds stated for Proscan are thus directly comparable to those figures quoted for other scanners listed in the comparative scanner performance table which appears this month on page 16.*

ROGUES' GALLERY

A Walk on the Wild Side - The UK's 'Least Wanted' List

This is *VB's* first published survey of actual virus attacks 'in the field'. The figures are necessarily distorted to some extent by local conditions and it is difficult to draw any firm conclusions from the virus types reported. Some other virus reports which just missed the 'top ten' list, were Italian (Ping-Pong), Disk Killer (Ogre), Vienna and Traceback.

Preliminary analysis of the list indicates that the older viruses (New Zealand, Cascade, Jerusalem, 4K and Vacsina) achieved a widespread penetration *before* the general introduction of effective virus detection methods. It may therefore be some time before there is any noticeable change in their reported penetration.

Four of the viruses in the *aide-memoire* which follows are either boot sector viruses or contain some boot infection element. These are generally more difficult to disinfect. Unsuspected infection on diskette is liable to result in problems of recurring infection despite the fact that the interchange of diskettes may not be prevalent. Eight viruses in the list are parasitic and the transfer and exchange of program files will certainly facilitate their propagation. On the other hand, the detection and identification of parasitic viruses, even those which use quite sophisticated encryption techniques, has become quite effective and this may eventually reduce the incidence of the older parasitic viruses in the list.

New Zealand (Stoned, Marijuana)

This virus has been reported extensively in the UK and Europe, the Far East, the US and Australasia. It was first reported in late 1987 in New Zealand and has since spread worldwide via the Far East.

It is a primitive boot sector virus which infects the Master Boot Sector of fixed disks as well as the boot sector of diskettes. The intention was to make a message 'Your PC is now Stoned' appear on every eighth reboot, but hackers have produced variants which now display a variety of effects including the formatting of hard disks. This is an instance of a so-called 'benign' virus causing immense damage and inconvenience to the global computing community.

The original virus stored a copy of the original Master Boot Sector on hard disks on Track 0, Head 0, Sector 2, but most later variants use either Sector 6 or Sector 7. Disinfection can usually be achieved by locating the original Master Boot Sector and copying it back to its correct position in Sector 1. Infection of diskettes may cause corruption depending upon the size of the diskette concerned and when corruption does occur, recovery may be difficult. The original report will be found in *VB*, May 1990, p.8 and a disinfection procedure appears in *VB*, September 1990, p.9.

Cascade, 1701/1704

This is a parasitic, memory-resident virus and the original variants only infected .COM files (including COMMAND.COM). It was one of the first to use simple code encryption and was originally designed to avoid infecting 'true blue' IBM machines and only to trigger between October and December 1988. However, later variants contained alterations to the trigger conditions to enable its continued survival. The original trigger was stolen from an existing 'joke' program which made letters on a text screen appear to fall (or cascade) down the screen. This effect is very photogenic and has been the subject of much publicity.

Recovery of infected files may be possible, but the best way to clean a machine is to reboot the machine from a clean system disk, delete all infected files and replace them with copies from the original master software. The original report on this virus is in *VB*, September 1989, p.9.

Jerusalem (1813, PLO, Israeli)

This is the most extensively plagiarised virus and now forms one of the largest groups (at least 26 variants at the last count). It is a parasitic, memory-resident virus which typically appends its code to .EXE files and prepends it to .COM files.

The original version had a trigger routine which deleted program files on any Friday 13th, but variants have abandoned this in favour of a wide range of conditions and triggers. Once again the best disinfection method is the complete replacement of infected files. A full report appeared in *VB*, July 1989, p.10 with subsidiary reports in August 1989 (p.10), October 1990 (p.8) and May 1991 (p.3).

Dark Avenger (Eddie, 2100)

These viruses emanate from a Bulgarian individual who has developed virus code obsessively since 1988. Various of his source code files are now circulating freely on Virus Exchange BBS systems and this has produced a rash of plagiarism. The original viruses are parasitic, resident and contain a variety of nasty trigger routines which damage both programs and data. Later versions have appeared which subvert anti-virus software and even target anti-virus add-on hardware boards. Once again, disinfection should be via the deletion/ replacement of infected files. The original report appeared in *VB*, February 1990, pp. 6-7. (A report on the related 2100 virus appeared in *VB*, August 1991, pp. 19-20.)

Virus Outbreaks in the United Kingdom

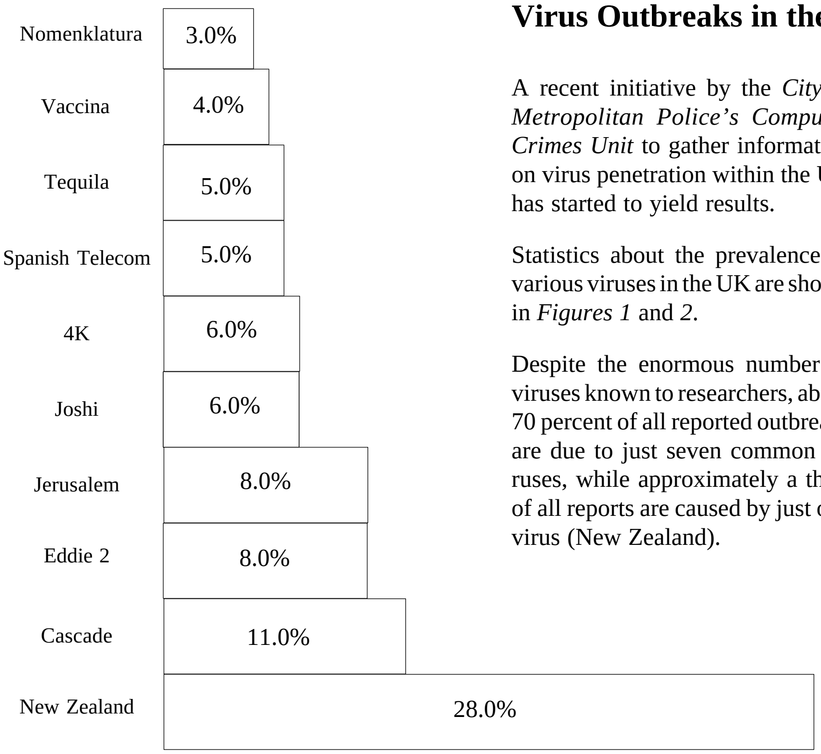


Figure 1. The most prevalent viruses in the UK. (*Bates Associates*)

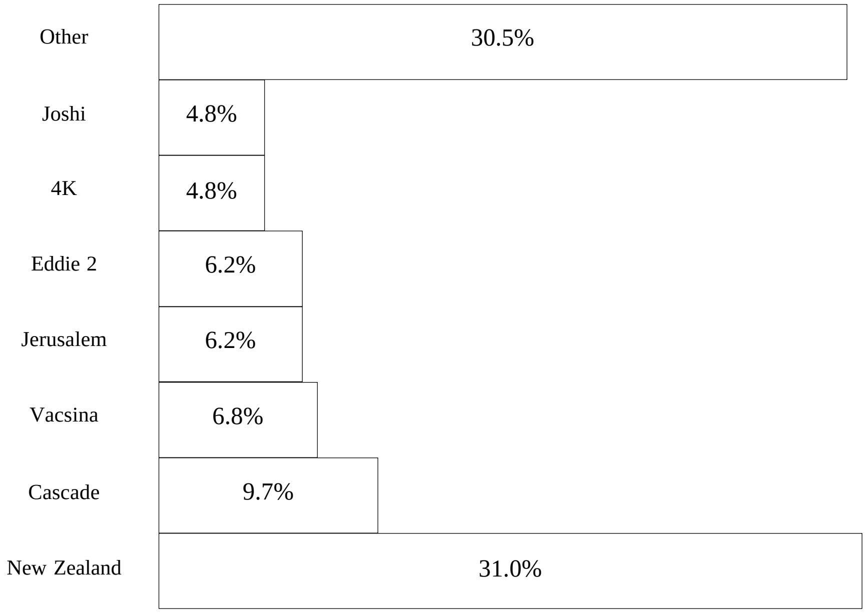


Figure 2. Virus reports to *Sophos* from 25th January - 22nd June 1991.

A recent initiative by the *City & Metropolitan Police's Computer Crimes Unit* to gather information on virus penetration within the UK has started to yield results.

Statistics about the prevalence of various viruses in the UK are shown in *Figures 1* and *2*.

Despite the enormous number of viruses known to researchers, about 70 percent of all reported outbreaks are due to just seven common viruses, while approximately a third of all reports are caused by just one virus (New Zealand).

In *Figure 2* the category 'other' accounts for a staggering 30.5 percent of all reported attacks - specimens in this category include Aircop, 777 Revenge, Form, Hallochen, Disk Killer, Invader, Italian, Keypress, Music Bug, Mystic, Nomenklatura, PcVrsDs, Printscreen, Slow, Spanish Telecom, Syslock, Tequila, 2100, Virdem, 1575 and Vienna.

The two developers which supplied statistics to *VB* concur broadly as to the identity of the 'top ten' viruses at large in the UK although, as should be expected, there are variations in their relative prevalence.

Some vendors of anti-virus software are beginning to differentiate between viruses 'at large' in the computing community and those believed to be research specimens. These measure have been forced upon them by the sheer weight of numbers. However, with the advent of Virus Exchange Bulletin Boards it is expected that the division between 'research' viruses and those reported at large will become less well-defined. It is thus apparent that any attempt to limit the number of viruses known to a particular package, carries the attendant risk of leaving its users unprotected against outbreaks of hitherto 'research' examples.

A careful watch is therefore necessary upon just which viruses are being reported by users. Although some long standing vendors have obviously had access to such information since the threat became manifest, no reliable statistics had become available until recently.

Now that the importance of such information is widely recognised, a more accurate assessment of computer virus penetration within the UK has become possible.

Joshi

This, like the New Zealand virus, is a Master Boot Sector infector. It was first reported in India in August 1990. The original version (no variants are currently known) triggers on January 5th and displays the message - *Type 'Happy Birthday Joshi' !* If these instructions are followed, operation continues normally. This is another so-called 'benign' virus which is not intended to cause damage. However, it is known to corrupt certain types of diskette and has caused access problems on some machines. The disinfection procedure is similar to that described for the New Zealand virus except that with Joshi, the original Master Boot Sector is stored on Track 0, Head 0, Sector 10. A report was published in *VB*, December 1990, pp. 17-18.)

4K (Frodo, 100 Years)

This was one of the first true 'stealth' viruses in that the code included routines designed to 'hide' changes in the size and content of infected files. It also contained an unusual mechanism to connect to and monitor the operating system services. The original trigger routine was designed to display the message 'Frodo Lives' at boot time but a bug in the code prevented this from operating as intended. Other trigger routines have been noted in some variants. Corruption has been reported occasionally due to the virus manipulating the disk cluster allocation mechanism and its propensity for infecting data files where the last three letters of the file specification total 223 or 225. Disinfection is by the usual deletion/replacement technique for parasitic, resident virus code but victims should be aware that data files may also have been corrupted. Reports appeared in *VB*, May 1990, pp. 10-11 and *VB*, November 1990, p.5.

Tequila

This recent virus was first reported in April 1991 and was written in Switzerland by people (since apprehended) reportedly associated with a shareware vendor. This presumably accounts for its rapid penetration into the European computing community. The virus uses self-modifying encryption and stealth tactics. It is a multi-partite virus, infecting both the Master Boot Sector and .EXE files. The trigger routine occurs at random and simply produces a crude Mandelbrot image on screen. Corruption will occur if the area that the virus uses to store its code (six consecutive sectors at the end of the active DOS partition) contains data.

Disinfection of the files can be accomplished by the usual deletion/replacement methods. Disinfection of the Master Boot Sector is more involved since the original boot sector must be located on the last track of the partition. This will typically be Sector 12 but may vary according to the actual configuration of the disk in question. Tequila was reported in *VB*, June 1991, pp. 16-17.

Spanish Telecom (Telefonica, Holocaust)

This is another multi-partite virus, first reported in late 1990 in Barcelona. This virus and its variants are known to have highly destructive trigger routines which may destroy all data on both first and (if there is one) second fixed disk drives. The trigger conditions are instigated by the boot portion of the code and invoked after a count of 400 reboots (whether by power-up or Ctrl-Alt-Del). The parasitic virus appends its code to COM files, avoiding COMMAND.COM and the IBM*.COM system files.

One of the unusual features of the original virus is that the boot infection does not contain the whole virus. This means that if a disk becomes infected with the boot section and infected files are removed, subsequent infection will continue only as a boot sector virus. This has led some researchers to insist that either the original report was incorrect, or there are two distinct viruses. Files should be disinfected by deleting and replacing them from clean masters under clean conditions. Disinfecting Master Boot Sector infections requires the same tactics as for the New Zealand virus - the original boot sector is stored at Track 0, Head 0, Sector 7. A report was published in *VB*, January 1991, pp. 22-24.

Vacsina (Yankee Doodle, TP Series)

Persistent reports of the Vacsina virus mainly refer to the variants which play 'Yankee Doodle Dandy' through the PC speaker. These are parasitic viruses which will infect executable files depending upon the contents of the header. File beginning with an E9H jump instruction, or an 'MZ' header identifier (EXE, OVL etc.) are targeted. No damage routines have yet been reported, the trigger seems mainly to involve the tune playing routine. Disinfection by replacement of infected files from clean master copies is recommended. No detailed report of Vacsina has yet been published in *Virus Bulletin*.

Nomenklatura

This is one of the most pernicious computer viruses so far developed. It is a primitive parasitic virus which employs neither stealth nor encryption routines and infects both COM and EXE files. The trigger routine is designed to cause maximum disruption by randomly interchanging allocated clusters within the disk structure. This can result in much more disruption than even a complete disk format since users may be tempted to continue operation after replacing infected files. Once a machine has been infected by this virus, the safest course is to completely reformat the hard disk and rebuild the system from scratch using clean software masters and data backups. However, if the virus is suspected of having infected the PC for some time, the backups may be corrupted. A report appeared in *VB*, December 1990, pp. 19-20.

SCANNER UPDATE

Mark Hamilton

Congratulations...

Congratulations all round this month. All the companies upon which we regularly report have brought out updates (or a signature file in the case of *Central Point*) to their scanners. The standard of all the products has improved immensely. Two suppliers, both from the United States, deserve special plaudits: *Symantec Inc.* and *Microcom Software Division.*

Not only has *Symantec* shipped version 1.5 of *Norton Anti-Virus*, it has also sent out the first update disk for the new version. The upgraded version is free to registered end-users. *Symantec UK* state that the company has produced four updates since the official launch of *Norton Anti-Virus* in December 1990. (One observation - the packaging for version 1.5 now bears a sticker proclaiming that NAV ‘Detects over 700 Viruses’ - but nowhere is any evidence to this effect produced. Companies would be well advised to avoid this numbers game - bland statements such as this are meaningless and confusing.)

Praise too, to *Microcom* for shipping version 1.2 of *Virex-PC* out to the user community. The company is now supplying free of charge a scanner (*VIRx*) via *Compuserve* (it is also available on other Bulletin Boards, particularly those in the ‘Virus Help’ chain). Both of *Microcom’s* scanning programs have performed especially well and are this month’s rising stars. *VIRx* has the ability to scan within certain compressed executable files. It correctly detects viruses packed with *LZEXE* and *PKLITE* but missed those compressed with *DIET*.

With more and more programs being distributed in compressed form, Ross Greenberg (the program’s developer) and *Microcom* have established something of a lead.

IBM has prepared a new version of *VIRSCAN*. It achieves a higher detection rating than the *S&S Toolkit* which the company also distributes. *IBM* is now routinely including *Virus Bulletin* search strings in its product, which probably accounts for its higher score.

Product	Version	Supplier	Updated?	Hard Disk Turbo	Hard Disk Secure	Diskette Turbo	Diskette Secure
CP Antivirus	1.0	Central Point	Yes	3:13	120:30	0:05	4:34
F-FCHK	1.16	Skulason	See Text	6:25	11:50	0:36	1:10
Findvirus	5.11	S&S/Ontrack	Yes	1:12	2:24	0:36	0:40
HTScan	1.15	Harry Thijsen	Yes	2:20	3:37	0:40	0:58
Norton Antivirus	1.5	Symantec	Yes	2:00	N/A	0:39	N/A
PC-Eye	2.0d	PC Enhancement	See Text	1:12	3:57	0:24	0:43
Scan	V80	McAfee Associates	Yes	3:51	7:05	1:02	1:34
Sweep	2.28	Sophos	Yes	3:45	5:35	0:42	0:55
TBScan	2.8	ESaSS	Yes	1:29	4:00	0:20	0:32
VIRx	1.6	Greenberg / Microcom	Yes	1:08	4:12	0:19	0:56
Virscan	2.1.2	IBM	Yes	3:40	4:50	1:06	1:26
Viscan	3.23	Bates / Total Control / Plus 5	Yes	3:19	3:25	0:19	0:25
Vi-Spy	7.0	RG Software	Yes	3:02	5:05	0:32	0:59
VPCScan	1.2	Microcom	Yes	3:18	3:23	0:19	1:25

RESULTS TABLE - SCANNING SPEEDS TESTS 1(i), 1(ii), 2(i), 2(ii) (See VB Testing Protocol, April 1991, pp. 6-7)

Product	Parasitic Turbo	Parasitic Secure	Boot Sector Turbo	Boot Sector Secure	Accuracy Turbo	Accuracy Secure
CP Antivirus	306	323	7	7	84.36%	88.95%
F-FCHK	357	357	6	6	97.32%	97.32%
Findvirus	353	353	7	7	97.03%	97.03%
HTScan	341	341	8	8	94.07%	94.07%
Norton Antivirus	315	N/A	7	N/A	86.79%	N/A
PC-Eye	349	350	8	8	95.71%	95.71%
Scan	344	344	8	8	94.88%	94.88%
Sweep	361	363	8	8	99.46%	100.00%
TBScan	332	332	8	8	91.64%	91.64%
VIRx	356	356	7	7	97.84%	97.84%
Virscan	353	355	8	8	97.30%	97.84%
Viscan	363	363	8	8	100.00%	100.00%
Vi-Spy	359	359	8	8	98.92%	98.92%
VPCScan	341	341	7	7	93.80%	93.80%

RESULTS TABLE - SCANNER ACCURACY TESTS 3/4 (See VB April 1991, pp. 6-7.)

Another scanner manufacturer who sent us an update which includes *VB* search strings is Harry Thijsen. He informs us that he does not include these strings with his standard product; if he did, his product would find nine more viruses. The *Virus Bulletin* strings, included in a separate data file, were not included in our tests.

Sophos' Sweep has started to report the outbreak of 'virus tops' - I find these 'top' indicators (whatever they are) confusing. During tests *Sweep* reported a virus infection ('Suriv 3.00 Top' and 'Jerusalem Top') in four known clean executable files. The company has been informed of these false-positive indications.

Fridrik Skulason is about to release version 2 of his shareware program, *FPROT*. Only beta-test versions of his new software were available while this update was in preparation and were disqualified accordingly. However, Skulason's new version should be available by the time *VB* goes to press.

Although *PC Enhancements* has updated its *PC Eye* to version 2.0f, it was unable to supply a stable version for review. The company acknowledges that it had a problem with the encoding of its signature file, which was discovered after the updates were sent out. The company says that it has now fixed the problem. I would advise any user who has this particular version with the signature file VIRSIG.VZF is dated prior to 14th August 1991 to revert to version 2.0d which is known to perform correctly. This review discounts the appearance of version 2.0f and repeats the results for version 2.0d which were published in the July edition of *Virus Bulletin*.

RG Software sent us *Vi-Spy* version 7 which, upon receipt, was still warm from the press (it arrived by *Federal Express* on 16 August). It has been updated to include search data for the Traveller virus which has recently hit organisations in Pennsylvania and Washington, DC. A new on-line help screen provides on-the-spot details of any virus which *Vi-Spy* detects.

Those UK readers unlucky enough to be hit by a virus are advised to report the matter to the police, preferably to the *Metropolitan and City Police's Computer Crimes Unit* (071-230 1176/7). *Total Control* has introduced an innovation in the latest version of its *VIS Utilities* - in the event of its detection software discovering a virus infection, a virus attack form (to *Computer Crimes Unit* specifications) is automatically generated. The company also includes printed versions of the form with its documentation. *Total Control* has appointed *Plus 5*, of Gaithersburg, Maryland, as its distributor for the United States and Canada.

TEST CONDITIONS

(See also Testing Protocol, *VB*, April 1991, pp 6-7)

A Compaq DeskPro 386/16 (a 16 MHz 386 ISA PC) with 6 Mb of RAM is used to test the speed of the scanner products. On the dedicated test hard drive partition (21 Mbytes) there are 786 files totalling 19 Megabytes. Of these, there are: 2 BIN, 112 COM, 223 EXE, 18 OV? and 33 SYS files. The scanners are invoked from a 3.5 inch diskette and the times taken include load time and, where applicable, automatic memory scans. Disk caching software was disabled. The diskette speed test is performed on a 5.25-inch 360 Kbyte disk (Microsoft C v5.1 Setup disk) containing 3 executable and 7 non-executable files occupying 354,747 bytes.

An Apricot 486/25 MCA-bussed PC is used for the scanner accuracy test. This PC is equipped with 16 Mb RAM and a 320 Mb SCSI hard disk and the virus library is held on its own dedicated disk partition.

VIRUS TEST-SET

The Virus Test Suite currently comprises 363 infected files containing, where appropriate, one COM and one EXE infection of the following viruses:

1067 (C); 1077 (C); 1226 (C); 1260 (C); 2480 (C); 3445 (C); 440 (C); 4K (CE); 5120 (CE); 555 (C); 789 (C); 800 (C); 8 Tunes (CE); Advent (C); Agiplan (C); Aids (C); Aids II (C); Akuku (C); Alabama (E); Ambulance (C); Amoeba (CE); Amstrad (C); Amstrad Cancer variant (C); Anthrax (CE); Anti-Pascal Family: AP-605 (C), AP-529 (C), AP-480 (C), AP-440 (C), AP-400 (C); Armagedon (C); Attention (C); Bebe (C); Best Wishes 1 (C); Best Wishes 2 (C); Blood; Black Monday (CE); Bulgarian 1600 (CE); Bulgarian 1600 v2 (C); Bulgarian 1600 v21 (C); Bulgarian 492 (C); Bulgarian 905 (C); Burger 1 (C); Burger 2 (C); Burger 3 (C); Burger-405 (C); Carioca (C); Cascade Family: (1) 01 (C), (1) 04 (C), (1) Y4 (C), Format (C); Casino (C); Casper (C); Christmas in Japan (C); Christmas Tree (C); Christmas Violator (C); Cookie (E); Crazy Eddie (C); Dark Avenger (CE); Dark Avenger-2100 (CE); Dark Avenger 3 (C); Datacrime Family: 1 (CE), 2 (CE), II (CE), IIB (E); Datalock (CE); Dbase (C); DBF Blank (CE); December 24 (E); Deicide (C); Destructor (CE); Devil’s Dance (C); Diamond A (CE); Diamond B (C); Dir (C); Diskjeb (CE); Do Nothing (C); Do Nothing 2 (C); Doom 2 (E); Dot Killer (C); Durban (CE); Dyslexia (C); Eddie-2 (CE); Evil (C); Faust (C); Fellowship (E); Fichv (C); Fish-6 (CE); Flash (CE); Flip (CE); Fu Manchu (CE); Gergana (C); Ghostballs (C); Guppy (C) Hallochen (E); Hybrid (C); Hymn (CE); Icelandic 1 (E); Icelandic 2 (E); Icelandic 3 (E); Int 13 (CE); Internal (E); Iraqui Warrior (C); Itavir (E); Jerusalem Family: 4th Black Friday (C), A204 (C), Anarkia (C), AntiScan (CE), B variant (CE), C variant (CE), GP1 (E), Groen Links (CE), Kylie (CE), Mendoza (C), PLO (C), PSQR (C), USA (C), Westwood (CE); Jocker (E); Jo-Jo (C); Joker-01 (C); July 13th (E); Justice (C); Kamikaze (E); Kemerovo (C); Kennedy (C); Keypress (CE): Lehigh (C); Leprosy (C); Leprosy B (CE); Liberty 1 (CE); Lovechild (C); Lozinsky (C); Machosoft (CE); MG (C); MG-1 (C); MG-2 (C); MG-3 (C); MG-4 (C); MGTU (C); Micro-128 (C); Minimal-45 (C); Mirror (E); Mix1 (E); Mix1-2 (E); Mix2 (E); MLTI (C); Monxla (C); Murphy-1 (CE); Murphy-2 (CE); Nina (C); Nomenklatura (CE); NTKC (C); Number of the Beast Family: A, B, C, D, E, F (C); Number 1 (C); Old Yankee 1 (E); Old Yankee 2 (E); Ontario (CE); Oropax (C); Parity (C); PcVrsDs (CE); Perfume (C); Phantom (C); Phoenix (C); Pixel Family: 1,2,3,5 (C); Plastique Family: AC-2900 (CE), AC-3012 (CE), AC-4096 (CE); Polish 217 (C); Polimer (C); Pretoria (C); Proud (C); Prudents (E); Raubkopie (E); Russian Group: 311 (C), 417 (C), 516 (C), 600 (C), 696 (C), 707 (C), 711 (C), 948 (CE), 1049 (CE), 2144 (CE), Mirror (C); Saddam (C); Scotts Valley (CE); Sentinel 1 (C); Shake (C); Slow (CE); South African 1 (C); South African 2 (C) South African 416 (C); Spanish (CE); Spanish Telecom (C); Staf (C); Stardot-801 (C); St. Petersburg (C); Subliminal (C); Sunday (CE); Suomi (C); Suriv 1.01 (C); Suriv 2.01 (E); Suriv 3.00 (CE); SVC v3.1 (CE); SVC v4.0 (CE); Sverdlov (CE); Svir (E); Sylvia (C); Syslock (C); Taiwan A (C); Taiwan B (C); Tenbyte (CE); Terror (C); Testvirus B (C); The Rat (E); Tiny (C); Tiny Family 1: T154 (C), T156 (C), T158 (C), T159 (C), T160 (C), T167 (C), T198 (C); Tiny Family 2: T133 (C), T134 (C), T138 (C), T143 (C); Traceback (CE); TUQ (C); Turbo 488 (C); Turbo Kukac (C); Twelve Tricks (C); Typo (C); V-1 (C); V2000 (C); V2P2 (C); V2P6 (C); Vacsina Family: TP04 (C), TP05 (C), TP06 (C), TP16 (C), TP23 (C), TP24 (C), TP25 (C); Vcomm (CE); VFSI (C); Victor (CE); Vienna Family: 1 (C), 2A (C), 2B (C), 3 (C), 4 (C), 5A (C), 5B (C), 6A (C), 6B (C), 582 (CE), 644 (C), 646 (C) 774 (C), 822 (C), Violator (C); Virdem Generic (C); Virdem 1 (C); Virdem 824 (C); Voronezh (CE); VP (C); Vriest (C); W13-A (C); W13-B (C); Whale (C); Wisconsin (C); Wolfman (E); XA-1 (1) (C); XA-1 (2) (C); Yankee Family: TP33 (CE), TP34 (CE), TP38 (CE), TP39 (C), TP41 (CE), TP42 (CE), TP44 (CE), TP45 (CE), TP46 (CE); Zero Bug (C); Zero Hunt (C).

In addition, the test-set comprises eight diskettes infected with the following boot sector viruses: Aircop; Brain; Disk Killer; Italian; Joshi; Korea; New Zealand 2; Spanish Telecom.

Two previous test-set specimens, namely Catman and Nazi have been removed pending further examination of their proven ability to replicate. In keeping with the policy to update the virus test-set each calendar quarter, additional virus samples will be added; details will be published in the October edition of *Virus Bulletin*.

BOOK REVIEW

Jim Bates

Burger's *Computer Viruses and Data Protection*

In spite of reasoned argument, heart-felt pleading and even threats, *Abacus* has now published the second edition of Ralf Burger's book on computer viruses in the United States and is on record as saying that it will be distributed in the United Kingdom 'shortly'. Thanks to Ray Glath of *RG Software Systems, VB* managed to obtain copies of this book prior to its release in the UK since rumours abounded concerning its contents. In the event, although some re-organisation is evident, the book contains little that is new and is essentially the same mish-mash of pseudo-technical jargon, irrelevant references, woolly reasoning and childish attempts to justify the virus phenonemon.

This edition has a new title - *Computer Viruses and Data Protection* and a new cover with the art-stencilled legend 'UNCLASSIFIED' together with a handwritten note stating 'Read this to find out all you need to know about Computer Viruses'. This alone is sufficient to indicate that the marketing strategy is to create the impression that the reader will become privy to sensitive or sensational information.

The Burger Interview

Before examining the contents, it is interesting to note some apparent confusion on the part of the author. In a recently published interview (August 1991) with an anti-virus vendor in the UK, Burger was described as 'a quiet, soberly-dressed German with longish but well-kept hair; he was wearing a business suit. His English was very good, but with occasional stumbles for an unfamiliar word.'

During the course of this interview Burger was asked how he could justify the publication of the Vienna virus disassembly in the original book. In his reported reply, he states, 'I did not at that time realise that people would misuse that disassembly so badly.' Such naivety is difficult to believe but it is just possible that it might be true.

Later in the interview, in reply to the question, 'If you had known in 1987 what you know now, would you have included that disassembly?', Burger replied, 'Definitely not.' However, in a completely new preface to the second edition of the book, dated June 1991, *only two months before this interview,* he writes, 'Some readers may believe that we should not have included the program code and examples published in this book. The reason that we included these examples is to show how easy it is to write a computer virus.'

A possible reason for these conflicting statements appears in a later portion of the interview when he reveals that he now writes and markets anti-virus software and wants to be accepted as a bona-fide researcher because: 'I need access to virus code and disassemblies for my anti-virus software'. This is breathtaking duplicity and one can only express pity for the interviewers for their gullibility in believing this man.

Factual Inaccuracy

This new edition has grown from 282 to 348 pages but this is as much due to reformatting as it is to new information. The number of sections has been reduced from 15 to 13 and there has been some expansion of the text in most of the chapters. An indication of the lack of accuracy can be found in a statement in the first chapter which cites the AIDS Information Diskette incident. The statement begins by suggesting that 'more than 10,000 copies' of the disk were mailed to large firms 'in North America'. It goes on to indicate that anyone who loaded the program 'found that all data on their hard drives had been deleted.' This case is still sub-judice in the UK but the facts of its distribution and effects are well known around the world: none were mailed to the US and data was certainly not deleted. For someone claiming the authority that Burger does, to be so spectacularly wrong adequately reveals his lack of interest in verifiable facts.

> *"...attempts in some quarters to return Burger to the genuine research community should be treated with the same comtempt that he has shown computer users worldwide."*

The only really significant changes are to be found in the chapters dealing with virus examples and protection strategies. Even here, some glaring errors and misconceptions have been overlooked. The Vienna virus is described as, 'an extremely clever computer virus with such a complex manipulation task that it's beyond the scope of this book to calculate its full effects.' Mention is then made of the source code listing later in the book!

The remainder of this chapter has been expanded by the inclusion of 283 virus recognition listings. Each entry in this consists of four lines giving the virus name, infection target, size and a recognition signature. Some indication of just how useless this listing is can be gleaned from two entries: listed as

Dark Avenger 1 and Dark Avenger 2. Both are of length 1800 bytes, both infect COM and EXE files and both are described as 'very destructive'. The recognition sequence for Dark Avenger 1 contains just eight bytes and produced four false positive indications when tested on my own system.

A disturbing addition to this chapter is the 'authorised' use of the *McAfee Virus Characteristics* list. Quite apart from the inaccuracies that these lists contains, I find it inexplicable that *McAfee Associates* has allowed its name and work to be associated with this book. The chapter also repeats the insidious suggestion that viruses may be used to protect software from theft. Burger has once again missed the point that laws already exist to prevent such theft.

The chapter dealing with protection strategies has borne the most extensive rewrite, but even here the content is vague and uncoordinated. The chapter begins with a section listing ten 'easy steps you can follow to avoid contracting a computer virus.' Among these are fatuous admonitions to 'Make all of your COM and EXE files read-only' and 'NEVER boot from a floppy disk drive' (*never*?). Other suggestions include 'Do not loan an original program disk to other users', you should instead loan a copy of the program disk and then format it when it is returned! No mention is made of the legal implications of 'loaning' copies of proprietary software.

Two 'virus detection' programs listed in assembler source code are included. The first of these (which doesn't work) searches for a marker of 909090h as the first three bytes of a specified file. The second program 'checks for marker 31/30 minutes' in the file date/time stamp. Aside from the fact that it actually checks for a 62 second marker, no mention is made of the fact that many programs now use this marker in a mistaken attempt at self-protection from viruses. This is precisely the sort of inaccuracy which has resulted in such confusion among programmers and developers. The summary of this section blithely states, 'It's very difficult to discover virus programs by using search routines.' - this does rather invalidate the earlier listing of 283 recognition patterns.

Fatuous Suggestions

The remainder of the chapter purports to tell the reader how to protect his software. After a crude explanation of (very) basic backup procedures, the book seriously suggests: 'To minimize the threat of a computer virus, many users have resorted to developing their own software.' Those users without the necessary expertise can 'contract another person as the developer.' This is just one example of a multitude of fatuous suggestions and 'observations' made throughout the book.

Moving on to 'Protecting Your Data', users are advised to make their data easy to check by avoiding 'obscure data structures'. It is also suggested (apparently seriously) that renaming program files so that the ubiquitous COM and EXE extensions no longer exist, is 'very helpful in limiting damage and recognizing a virus.'. Obviously Burger has not yet encountered the viruses which infect on the LOAD and EXECUTE request and determine their actions by the presence or absence of the 'MZ' marker.

After further convoluted and often irrelevant sections concerning 'protection' on hard drives, BBSs, networks and even users(!!?) the chapter ends with two pages entitled 'What If You're Infected?'. This restates the obvious steps that are listed *ad nauseam* in *any* article about viruses - power down, isolate, reboot from a clean system, rebuild from masters etc. It also includes advice to backup data (after infection) while completely omitting mention of possible boot sector virus infection. All in all, this section is even *more* poorly presented than the remainder of the book. I could not shake off the feeling that it was simply included so that the author and publisher could say, 'Look, we're helping people to *fight* the virus threat', the result is worse than useless and in no way justifies any of the other chapters.

Source Code

The remainder of the book, including the original source code listings, is essentially unchanged. These include source code in a variety of languages including assembler and it is these which have caused such universal condemnation ever since the original publication. A concerned user would gain nothing from the listings, while a potential virus writer would find them *extremely* useful. As mentioned previously, the publisher has been informed in no uncertain terms that the inclusion of source code is considered thoroughly reprehensible by responsible sections of the computer industry. Legal opinion in the UK has indicated that there may be a case for action against the author/publisher/distributor if loss is sustained as a result of virus code or techniques traceable to this book.

In my review of the original English language edition of this book (*VB*, October 1989, p.19) I left no doubt that I found it inaccurate, distasteful and irresponsible. By his additions and alterations to this new edition, Burger has confirmed his position as a cynical, ill-informed and irresponsible individual who has no legitimate function in computing. His motives, his technical appreciation and his veracity are highly questionable and attempts in some quarters to return him to the genuine research community should be treated with the same contempt that he has shown for computer users worldwide.

Computer Viruses and Data Protection
Author: Ralf Burger
Price: US$ 19.95
ISBN 1-55755-123-5
Publisher: *Abacus*, 5370 52nd Street SE, Grand Rapids, MI 49512, USA.

PRODUCT REVIEW

Dr. Keith Jackson

Thunderbyte

Thunderbyte comprises a half length plug-in card (designed for the ISA or EISA bus) and associated software. It provides facilities to: scan files to see whether they are virus infected; detect virus execution dynamically; create a correct directory listing (even on a PC where a virus is returning false information); and construct a ‘rescue’ disk to be used when serious damage has been caused by a virus. The software is provided on both 3.5 inch and 5.25 inch media.

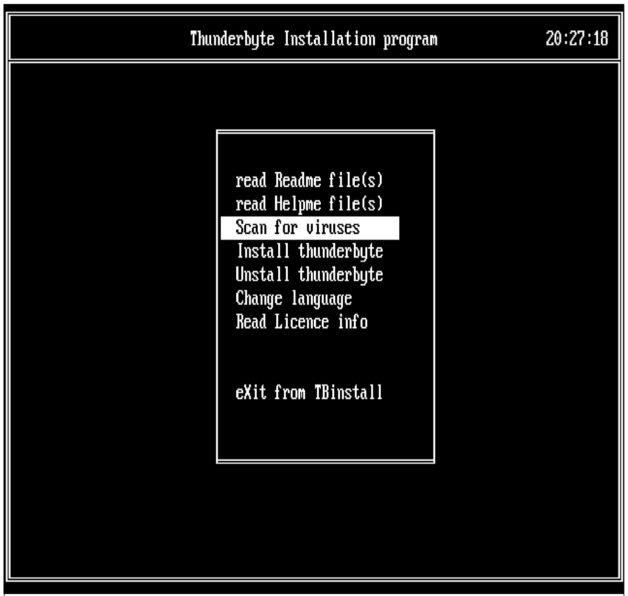
Thunderbyte claims to provide protection against viruses using hardware so as to ensure that it can take control of the PC before the operating system is loaded, and to detect virus activity as it occurs. The data cable to the hard disk can be routed via the *Thunderbyte* plug-in card, allowing *Thunderbyte* to control the hard disk at a low level. This re-routing of the hard disk cable is not mandatory, and in my case was impossible as the PC used for testing viruses only uses a hardcard, and the data cables are not accessible. *Thunderbyte* states that it will operate under Windows 3.0, and fully co-operate with networks. I did not test either of these claims.

Installation

The average user should readily be capable of installing *Thunderbyte*. First the software is installed in any desired subdirectory using the installation program provided, then the DIP switches on the plug-in card are set (as advised by the installation program). Finally the plug-in card is inserted into any available slot. The software installation program was very easy to use, and provides extensive on-screen help in a language chosen by the user. Currently this choice is restricted to English, Dutch and German, though the developers hope to extend this in the near future. The installation program was particularly helpful in drawing a picture on the screen of how it thought the DIP switches should be set on the plug-in card.

The *Thunderbyte* plug-in card is easy to install; in some ways too easy. It only requires a half length slot, but it is not polarised to prevent insertion the wrong way round. The documentation, and the card itself, warn the user against doing this. I did not test what happens if the *Thunderbyte* card is inserted the wrong way round for fear of causing physical damage. After the plug-in card has been inserted, *Thunderbyte* will take control the next time that the PC is booted.

Two ‘features’ of the *Thunderbyte* controlled boot process annoyed me. First, the plug-in card always takes control before MS-DOS is allowed to boot. This can be used to ensure that the PC is in a ‘clean’ condition before MS-DOS is loaded. Normally a PC first carries out its own tests (Power On Self-Test or POST) to check that the hardware is functioning correctly, then it loads the operating system (MS-DOS) from disk. This takes 29 seconds on my test PC when booting from floppy disk. The extra tests introduced by *Thunderbyte* increased the overall boot time to 2 minutes 9 seconds; more than a factor of four.



Installing *Thunderbyte* is simplicity itself. Note the ‘change language’ option - languages currently available are Dutch, English and German.

During the tests carried out for this review, when *Thunderbyte* intervened and prevented what it thought was a virus infected program from carrying out an action, the PC was often locked up so thoroughly that a power down was required. Waiting for over two minutes every time this happened proved to be tedious in the extreme.

When *Thunderbyte* has finished its startup tests, it displays a copyright message and says “Press Ctrl-Space for options window”. This did not work. I tried many other key combinations, but nothing would persuade *Thunderbyte* to display its options window. Therefore this review only investigates the default *Thunderbyte* configuration.

Documentation

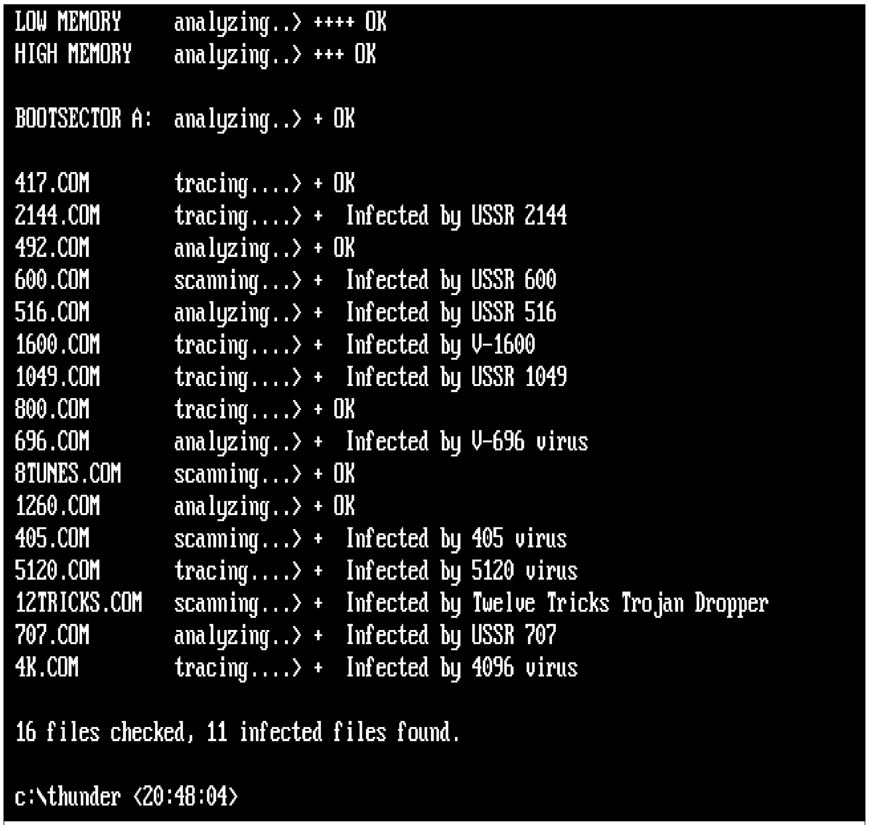
Thunderbyte comes with a 37 page A5 booklet which is well written, and avoids the grandiose claims seen in many anti-virus products. It does not have an index (no apologies for dragging this old chestnut up yet again - indexes help *users* to locate information). The booklet explains what a virus is, what *Thunderbyte* does, how to configure and install *Thunderbyte*, and what warning messages may appear. As an addendum to the main manual, many files are provided on disk in the form of README files. Development of *Thunderbyte* has obviously been proceeding apace, as these files are now so extensive as to warrant inclusion within the main manual.

Virus-Specific Scanning

As the scanning program provided with *Thunderbyte* (*TBSCAN*) is included in the *VB* comparative review of scanner programs, I will not dwell on it. Suffice it to say that *TBSCAN* correctly detected a virus in 87 percent of infected programs. This is within a couple of percentage points of the results quoted in the *VB* comparative review of scanner programs, to which the reader is referred for more detailed information about *TBSCAN*. (See *Scanner Update*, pp. 19-21.)

Dynamic Detection

Whenever the *Thunderbyte* plug-in card detects what it perceives as virus related activity, it intervenes and pops up a small message window in the middle of the screen. The on-screen messages are clear and unambiguous. They offer a one-sentence summary of what *Thunderbyte* has detected, and a few options to be followed. The preferred option is usually clearly indicated. The messages that were produced during testing were varied and included warnings that an interrupt vector was being changed; a program was being modified; direct disk writes were occurring; COMMAND.COM was being modified; a program was trying to go memory-resident; and a 'stealth type' virus had been prevented.



```
LOW MEMORY    analyzing..> ++++ OK
HIGH MEMORY   analyzing..> +++ OK

BOOTSECTOR A: analyzing..> + OK

417.COM       tracing....> + OK
2144.COM      tracing....> +  Infected by USSR 2144
492.COM       analyzing..> + OK
600.COM       scanning...> +  Infected by USSR 600
516.COM       analyzing..> +  Infected by USSR 516
1600.COM      tracing....> +  Infected by V-1600
1049.COM      tracing....> +  Infected by USSR 1049
800.COM       tracing....> + OK
696.COM       analyzing..> +  Infected by V-696 virus
8TUNES.COM    scanning...> + OK
1260.COM      analyzing..> + OK
405.COM       scanning...> +  Infected by 405 virus
5120.COM      tracing....> +  Infected by 5120 virus
12TRICKS.COM  scanning...> +  Infected by Twelve Tricks Trojan Dropper
707.COM       analyzing..> +  Infected by USSR 707
4K.COM        tracing....> +  Infected by 4096 virus

16 files checked, 11 infected files found.

c:\thunder <20:48:04>
```

A graphic illustration of the shortcomings of virus-specific detection. *TBSCAN* passing five virus infected .COM files as 'OK' during a routine scan. In recognition of this, the *Thunderbyte* hardware is an attempt to provide generic defence against computer viruses.

When the program infected with the Ambulance virus was executed, *Thunderbyte* did not activate even though the virus was obviously present from the presence of an ambulance scuttling across the bottom of the screen while wailing its siren. This virus is thought to have no effects other than producing the moving ambulance, so what was there for *Thunderbyte* to detect? [*apart from the virus going memory-resident? Ed*.]. The distinction between a virus infected program and an uninfected computer program is often difficult to ascertain.

Fatal Error

In two cases (the Attention virus, and the Lovechild virus) execution was abruptly terminated with the message "Fatal error, damage in *Thunderbyte* RAM area". One of the README files provided with *Thunderbyte* discusses this point, and offers various palliatives, but it is a graphic demonstration that without its own processor and/or secure memory area, a plug-in card will always suffer from this problem. There is nothing in the PC architecture to prevent a program (or virus) writing to any part of memory that it so chooses. The effects are often catastrophic, which may well be exactly what a virus author desires.

For 69 of the 183 infected test programs, *Thunderbyte* did not activate. However in 25 of these cases DOS complained about an attempt to write to a write protected disk. The instances where a virus was thwarted by a write-protect tab on drive C: (a floppy disk drive on my test PC) would presumably be intercepted if *Thunderbyte* was installed so that the hard disk data cables were connected via the *Thunderbyte* card.

Only 31 of the 183 test samples (17 percent) resulted in MS-DOS automatically regaining control after execution of a virus infected program. Even in these cases *Thunderbyte* may well not detect virus activity for the simple reason that during testing the virus was not actually doing anything. For instance many viruses have a specific trigger date which must be within prescribed limits before activation occurs. Therefore the figure of 17% quoted above for failure to detect virus activity is almost certainly a worst case value, and probably underestimates *Thunderbyte*'s detection capabilities. Barring going through every virus and figuring out exactly what it should do, I see no easy way to improve on the current results, and as *Thunderbyte* emerges from these tests with its head held high, I would contend that such an exercise is pointless.

In 28 out of the 181 parasitic virus test samples used, *Thunderbyte* detected virus activity while the virus was merely being copied from one disk to another. In all cases *Thunderbyte* reported that an attempt was being made to set the timestamp associated with the file to an illegal value (e.g. 62 seconds). This method is used by some viruses to mark a file as being already infected, thereby preventing multiple infection of single files. However, the MS-DOS copying utility is simplistic about such matters as it passes the illegal date to the newly created file. As no other error was detected in any of 181 copying tests, I can only assume that *Thunderbyte* does not check the content of a file while it is being copied. This conclusion is consistent with the low overhead imposed by *Thunderbyte* (see below).

Imposed Overhead

As the *Thunderbyte* plug-in card monitors program execution, and does not have its own processor, it must use the processor within the PC to execute the *Thunderbyte* software. Inevitably this must impose an overhead on the PC, and slow down execution somewhat. I carried out tests designed to measure the size of this overhead, involving multiple file copying, and timing of program execution. In no case could I detect any measurable overhead from having *Thunderbyte* installed. The overhead cannot be zero otherwise *Thunderbyte* would not be doing any checking, however it does seem to be so small that for practical purposes it can be ignored.

Concluding Thoughts

I should point out that the *Thunderbyte* card used in this review was actually the second card tested. The first card did nothing other than provide an on-screen message 'RAM parity error' immediately after applying power. I mention this as a cautionary tale - I assumed that this message was being issued by my own computer during its Power On Self Test sequence. This was confirmed by the the *Thunderbyte* developers in Holland. It appears that the message was issued by my system's BIOS failing to execute the corrupt *Thunderbyte* extension ROM (the checksum of which was incorrect). The replacement card worked correctly.

Overall I quite liked the product. It is undergoing further development (always a good sign), as the software provided with the replacement card was not the same as the original software provided. With the exception of the extremely long extension to the boot time, I found *Thunderbyte* to be fairly unobtrusive. *Thunderbyte* does not impose a large overhead on program execution, and given the task it sets itself of detecting virus activity dynamically it proved to be reasonable at preventing viruses from carrying out their dastardly deeds. *Thunderbyte* is not perfect in this respect, but given that there is no unique way of discriminating a virus infected program from any other computer program, no dynamic detector of virus activity will ever be 100 percent effective. I feel quite strongly that the hardware should be modified in some way so that incorrect insertion of the card is impossible. Even if *Thunderbyte* is not itself damaged by incorrectly insertion, such an action could conceivably damage other components.

Technical Details

Product: *Thunderbyte*

Developer: ESaSS B.V., P.O.Box 1380, 6501 BJ Nijmegen, The Netherlands, Tel (support BBS): +31 (85) 212 395

UK Vendor: Tekware Ltd., The Barclay Centre, 127A Worcester Road, Hagley, West Midlands DY9 0NW, England, Tel: +44 (562) 882125, Fax: +44 (562) 884855

Availability: The types of PC on which *Thunderbyte* will operate successfully are not specified in the documentation, but an ISA or EISA bus is mandatory for the *Thunderbyte* plug-in card.

Version Evaluated: 2.1

Serial Number: TB050391/05

Price: The scanning part of *Thunderbyte* (*TBSCAN*) is shareware (with no payment requested for either personal use or company use), but the *Thunderbyte* card itself costs £99.00 excluding VAT and delivery.

Hardware Used: An ITT XTRA (a PC clone) with a 4.77 MHz 8088 processor, one 3.5 inch (720K) floppy disk drive, and two 5.25 inch floppy disk drives running under MS-DOS v3.30.

Virus Test-Set: This is a suite of 114 unique viruses (according to the virus naming convention employed by *VB*), spread across 183 individual virus samples. It comprises two boot sector viruses (Brain and Italian), and 112 parasitic viruses. There is more than one example of many of the viruses, ranging up to 12 different variants in the case of the Tiny virus. Where more than one variant of a virus is available, the number of examples of each virus is shown in brackets.

1049, 1260, 12 TRICKS, 1600, 2144 (2), 405, 417, 492, 4K (2), 5120, 516, 600, 696, 707, 800, 8 TUNES, 905, 948, AIDS, AIDS II, Alabama, Ambulance, Amoeba (2), Amstrad (2), Anthrax (2), Anti- Pascal (5), Armagedon, Attention, Bebe, Blood, Burger (3), Cascade (2), Casper, Dark Avenger, Datacrime, Datacrime II (2), December 24th, Destructor, Diamond (2), Dir, Diskjeb, Dot Killer, Durban, Eddie 2, Fellowship, Fish 6 (2), Flash, Flip (2), Fu Manchu (2), Hymn (2), Icelandic (3), Internal, Itavir, Jerusalem (2), Jocker, Jo-Jo, July 13th, Kamikaze, Kemerovo, Kennedy, Keypress (2), Lehigh, Liberty (2), LoveChild, Lozinsky, MIX1 (2), MLTI, Monxla, Murphy (2), Nina, Number of the Beast (5), Oropax, Parity, Perfume, Piter, Polish 217, Pretoria, Prudents, Rat, Shake, Slow, Subliminal, Sunday (2), Suomi, Suriv 1.01, Suriv 2.01, SVC (2), Sverdlov (2), Svir, Sylvia, Taiwan (2), Terror, Tiny (12), Traceback (2), TUQ, Turbo 488, Typo, Vacsina (8), Vcomm (2), VFSI, Victor, Vienna (8), Violator, Virus-101 (2), Virus-90, Voronezh (2), VP, V-1, W13 (2), Whale, Yankee (7), Zero Bug.

END-NOTES & NEWS

The second edition of the *PC Virus Control Handbook* by Robert V. Jacobson is now available. The book provides a thorough introduction to PC viruses, their detection and effective disinfection. The book costs US$ 24.95 and is available from bookshops or directly from *Miller Freeman Publications*, Book Division, 500 Howard Street, San Francisco, CA 94105, USA.

IBC Technical Services is running a seminar on **Managing Network Security in the '90s** in London on September 13th. Tel 071 236 4080.

IBM is running a **Virus Master Class** on September 16th (Manchester, UK) and September 18th (Sudbury, UK). Contact *IBM Education Administration*, UK. Tel 0256 56144.

The *European Institute for Computer Anti-Virus Research* is holding a **two day seminar** in Brussels on September 24-25th. Contact Guenther Musstopf (Tel +49 40 6932033) or Dirk Giroulle (Tel +32 3 2316308).

Sophos continues a series of **computer virus workshops**. Introductory and advanced course are provided. The next available courses are on October 9th and 10th and take place in Oxford, UK. Tel 0235 559933.

Frost and Sullivan is holding a three-day course on **Microcomputer Security and Computer Viruses** in London, October 7th-9th. Tel 071 730 3438.

S&S Ltd is holding a **seminar on the virus threat** in Buckinghamshire, UK, on October 9th-10th. Tel 0442 877877.

Professional Development Associates is holding a two-day seminar on **NetWare Security** in London on October 17-18th.

An interesting letter addressed to our very own Petra Deffield [sic] arrived at the *VB* offices recently. Basit Farooq Alvi, System Analyst at *Brain Computer Services*, Lahore, Pakistan had read about the *Virus Bulletin Conference* in *Byte* magazine and wanted further information. Basit, the author of the original Brain virus, seems to be doing well for himself (he even provides a mobile phone and fax number) - 'We deal in Intel 8088, 8086, 80286, 80386SX, 80386 80486 based computers' runs the footnote of *Brain Computer Services'* embossed letter. Strangely, virus production and dissemination doesn't appear in the extensive list of other services the company provides.

VIRUS BULLETIN

Subscription price for 1 year (12 issues) including first-class/airmail delivery:

UK £195, Europe £225, International £245 (US$395)

Editorial enquiries, subscription enquiries, orders and payments:

Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Abingdon,
OX14 3YS, England

Tel (0235) 555139, International Tel (+44) 235 555139
Fax (0235) 559935, International Fax (+44) 235 559935

US subscriptions only:

June Jordan, *Virus Bulletin*, 590 Danbury Road, Ridgefield, CT 06877, USA
Tel 203 431 8720, Fax 203 431 8165

No responsibility is assumed by the Publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.